Report 6 - Stochastic methods for finance

Roben Bhatti - 2091187 - Physics of data

May 12, 2023

Abstract

In this report we simulate 100 trajectories of Geometric Brownian Motion in a Black&Scholes market with Monte Carlo methods and then we build a pricer for Vanilla and Asian options. The results will show compatible values with Black&Scholes formulas.

1 Introduction

Monte Carlo simulation uses the risk-neutral valuation result. We sample paths to obtain the expected payoff in a risk-neutral world and then discount this payoff at the risk-free rate. Consider a derivative dependent on a single market variable S that provides a payoff at time T. Assuming that interest rates are constant, we can value the derivative as follows:

1. Sample a random path for S in a risk-neutral world.
2. Calculate the payoff from the derivative.
3. Repeat steps 1 and 2 to get many sample values of the payoff from the derivative in a risk-neutral world.
4. Calculate the mean of the sample payoffs to get an estimate of the expected payoff in a risk-neutral world.
5. Discount this expected payoff at the risk-free rate to get an estimate of the value of the derivative.

The process followed by the underlying market variable in a risk-neutral world is:

$$dS = \mu S dt + \sigma S dz \tag{1}$$

Where dz is a Wiener process, μ is the expected return in a risk-neutral world, and σ is the volatility. To simulate the path followed by S, we can divide the life of the derivative into N short intervals of length dt and approximate equation 1 as:

$$S(t + \Delta t) - S(t) = \mu S(t) dt + \sigma S(t) \epsilon \sqrt{\Delta t} \tag{2}$$

where S(t) denotes the value of S at time t, ϵ is a random sample from a normal distribution with mean zero and standard deviation of 1.0, if S is the price of a non dividend paying stock then μ=r. This enables the value of S at time Δt to be calculated from the initial value of S, the value at time 2 Δt to be calculated from the value at time Δt, and so on. One simulation trial involves constructing a complete path for S using N random samples from a normal distribution. Using Ito's lemma and taking μ, σ constant it is possible to write the equation:

$$S(T) = S(0)exp\left[(\mu - \frac{\sigma^2}{2})T + \sigma\epsilon\sqrt{T}\right] \tag{3}$$

The key advantage of Monte Carlo simulation is that it can be used when the payff depends on the path followed by the underlying variable S as well as when it depends only on the final value of S. (For example, it can be used when payoffs depend on the average value of S between time 0 and time T). Payoffs can occur at several times during the life of the derivative rather than all at the end. Any stochastic process for S can be accommodated. The drawbacks of Monte Carlo simulation are that it is computationally very time consuming and cannot easily handle situations where there are early exercise opportunities.

1.1 Asian options

The payoff of an Asian option is based on the difference between an asset's average price over a given time period, and a fixed price called the strike price. Asian options are popular because they tend to have lower volatility than options whose payoffs are based purely on a single price point. It is also harder for big traders to manipulate an average price over an extended period than a single price, so Asian options offer further protection against risk. One disadvantage of Asian options is that their prices are very hard to compute using standard techniques, but this is a more achievable using Monte Carlo simulations. The payoffs for Call and Put are:

$$C(T) = max(A(0,T) - K, 0) \qquad P(T) = max(K - A(0,T), 0) \tag{4}$$

where A(0,T) is the average that can be computed in different ways according to the model. For our purpose we consider the standard mean.

2 Methods & Results

2.1 Geometric Brownian Motion

First we simulate a geometric brownian motion with constant drift μ and volatility σ using equation 1 sampling dz from a normal distribution and setting the following parameters:

- S=100
- K=00
- T=1 year
- R=1%

In fig. 1 there is the plot with 100 simulations of the whole paths.

2.2 Vanilla options

It is possible to implement a pricer for call and put options based on Monte carlo simulations using code from 3. The Monte Carlo pricing is done in two ways: one step and multi step simulation. For the multi step we use the paths generated above and for each path we evaluate the payoff:

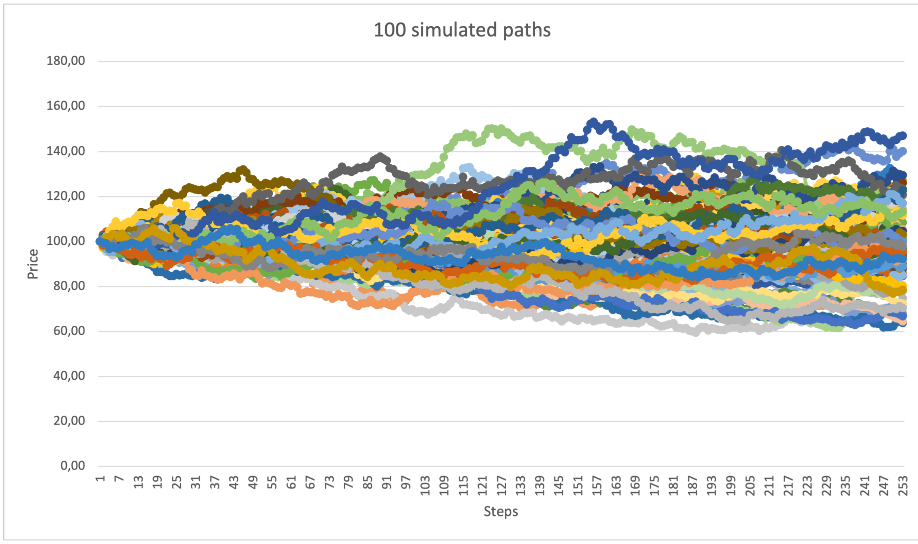

Figure 1: GBM simulation of 100 paths.

C(T)=max(S_T-K,0) and P(T)=max(K-S_T,0) and then take the mean because each path is equally probable. For the one step we set N=1 and we generate 1000 simulation of the final price with equation 3 and then we take the mean to price the option. So in this case we don't simulate the whole path. The values are reported in table 1. As a benchmark we computed also the option prices with the Black Scholes model and we can see that the 1 step MC simulation is closer to the B&S value. this behavior is expected since the multi step simulation involves more discretization.

2.3 Asian option

Asian options are path dependent options so to price them we can modify slightly the code above in order to keep track of the average stock price until maturity and by implementing the corresponding payoff. Of course we can price them only by using a multi step Monte Carlo simulation. The code is showed below and the prices in Table 1. In general we can say that their prices are lower than vanilla options, This is because Asian options are calculated based on the average price of the underlying asset over a period of time, which smooths out some of the short-term fluctuations in the asset price.

Listing 1: VBA modified code for **asian option pricing**

```



Function Asian(S0 As Double, K As Double, r As Double, sigma As Double,
T As Double, N As Long, paths As Long, Optional Call_Put
As Integer = 1) As Double

    Dim dt As Double, payoff As Double, S As Double, S_asian() As Double
    ReDim S_asian(1 To N)
    dt = T / N
    payoff = 0
    Dim i As Long, j As Long
```

```
    For i = 1 To paths
        S = S0
        For j = 1 To N
            S = S + r * S * dt + sigma * S * Sqr(dt) *
            Application.NormInv(Rnd(), 0, 1)
            S_asian(j) = S
        Next j
        If Call_Put = 1 Then
            payoff = payoff + Exp(-r * T) *
            WorksheetFunction.Max(Application.Average(S_asian) - K, 0)
        ElseIf Call_Put = -1 Then
            payoff = payoff + Exp(-r * T) *
            WorksheetFunction.Max(K - Application.Average(S_asian), 0)
        End If
    Next i
    Asian = payoff / paths
End Function
```

	vanilla call	**vanilla put**	**asian call**	**asian put**
1 step MC	8,12	7,64	x	x
multi step MC	8,74	7,22	5,35	4,21
Black Sholes	8,48	7,38	x	x

Table 1: Comparison with different methods and options.

3 Conclusions

In this report we built an option pricer for vanilla and asian types using Monte Carlo methods. We saw that with vanilla option this method gives comparable results with respect to the Black Scholes model. We have computed the pricing with different steps of MC simulation and found out that the multi step one tends to overprice the option with respect to its Black Scholes value. For the Asian options we see that in general they are priced less than the vanilla ones this is because of their less volatility. Also accuracy of the MC simulations can increase by setting $10^4 \sim 10^6$ simulations but the software limitation with excel does not allow this, in any case for our purpose it's more than enough to have a thousand of simulations.

Appendix

VBA codes are taken from these links:

- GBM: `https://investexcel.net/geometric-brownian-motion-excel/`
- vanilla option pricing: `https://www.spreadsheetweb.com/how-to-calculate-option-pricing\-using-monte-carlo-simulations-in-excel/`
- Black Scholes : `https://sites.google.com/view/vinegarhill-financelabs/black-scholes\black-scholes-merton`